## IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method of providing concurrent access to data on cross-computing environments, comprising:

receiving a first request for the data from a first computing environment;

receiving a second request for the data from a second computing environment, wherein the second computing environment is different from the first computing environment;

identifying a first message file to service the requests; and

concurrently providing access to the first message file to both the first computing environment and the second computing environment in order to load the data as directed by the first request from the first computing environment and the second request from the second computing environment, wherein the first message file is for dynamically rendering and translating the data into a desired language associated with the first and second requests and wherein at least a portion of the message file is dynamically resolved in response to the first computing environment, and wherein the portion is different from another portion that is dynamically resolved from the message file in response to the second computing environment, and wherein the desired language is related to a language used in a particular country having a particular dialect.

2. (Original) The method of claim 1, further comprising loading the first message file into a memory before providing the message file to the requests.

3. (Original) The method of claim 1, further comprising loading the first message file into a data store before providing the message file to the requests.

4. (Original) The method of claim 1, wherein in identifying the first message file, the first message file is identified with a first language associated with the requests.

5. (Original) The method of claim 1, wherein in providing the first message file, the first message file is represented in a generic file format.

6. (Original) The method of claim 1, further comprising:

loading the first message file into a memory;

receiving a third request from a third computing environment for the data, wherein the third request requires a second message file associated with a different language than required by the first request and the second request;

identifying the second message file;

loading the second message file into the memory concurrently with the loaded first message file; and

concurrently providing access to the second message file in order to load the data as required by the third request from the third computing environment while access is provided to the first message file to satisfy the first request and the second request.

7. (Original) The method of claim 6, wherein in receiving the third request, the third computing environment is the same as the first computing environment or the second computing environment.

8. (Previously Presented) The method of claim 6, wherein in receiving the third request, the third computing environment is the different from the first computing environment and the second computing environment.

9. (Currently Amended) A method of providing access to data in cross-languages and on cross-computing environments, comprising:

receiving a first request for the data, wherein the first request requires the data to be provided in a first language on a first computing environment;

receiving a second request for the data, wherein the second request requires the data to be provided in a second language on a second computing environment;

identifying a first message file of the data for the first language;

identifying a second message file of the data for the second language; and

concurrently providing the first message file and the second message file to service the first request and the second request in the first computing environment and the second computing environment, and wherein the first and second message files are for dynamically rendering and translating the data into different languages and wherein at least a portion of the first and second message files are dynamically resolved in response to the first computing environment and the second computing environment, respectively, and wherein the different languages relate to different languages associated with different countries or different dialects within a same country.

10- 15. (Cancelled).

16. (Currently Amended) A system to provide access to data in cross-language formats and on cross computing environments, comprising:

a first request for the data from a first computing environment;

a second request for the data from a second computing environment;

a first message file;

a second message file; and

a set of language manager executable instructions operable to identify the first message file and load it into a first data structure representing the data in a first language, the set of language manager executable instructions also is further operable to identify the second message file and load it into a second data structure representing the data in a second language, and wherein the first and second message files are for dynamically rendering and concurrently translating the data into the first language and the second language, and wherein at least a portion of the first message file and the second message file is dynamically resolved in response to the first computing environment and the second computing environment, respectively, and wherein

the first and second languages are associated with languages of different countries or different dialects for a same country.

17. (Original) The system of claim 16, wherein the first computing environment is different from the second computing environment and the first language is different from the second language.

18. (Original) The system of claim 16, wherein the set of language manager executable instructions loads the first message file and the second message file into a memory.

19. (Original) The system of claim 16, wherein the set of language manager executable instructions loads the first message file and the second message file into a data store.

20. (Original) The system of claim 16, wherein at least one of the requests is received over the Internet using a set of browser executable instructions.

21. (Original) The system of claim 16, wherein the set of language manager executable instructions retrieves the data from the first data structure and the second data structure to satisfy the first request and the second request.

22. (Original) The system of claim 16, further comprising a single application programming interface library providing an interface between the set of language manager executable instructions and the requests from the computing environments.

23. (Original) The system of claim 16, wherein the requests and the message files are represented in an extensible markup language format.

24-29. (Cancelled)